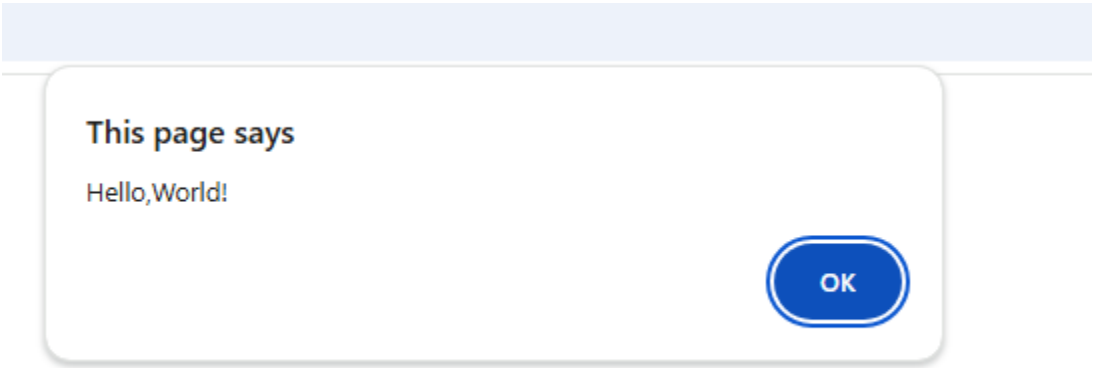
TASK 1:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        alert("Hello,World!");
    </script>
</html>
```

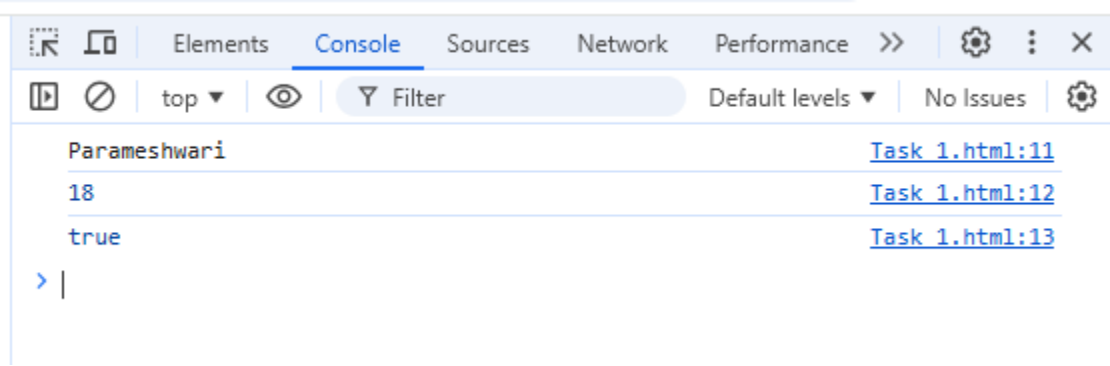
OUTPUT:



TASK 2:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        var name="Parameshwari";
        let age=18;
        let boolean=true;
        console.log(name);
        console.log(age);
        console.log(boolean);
    </script>
</html>
```

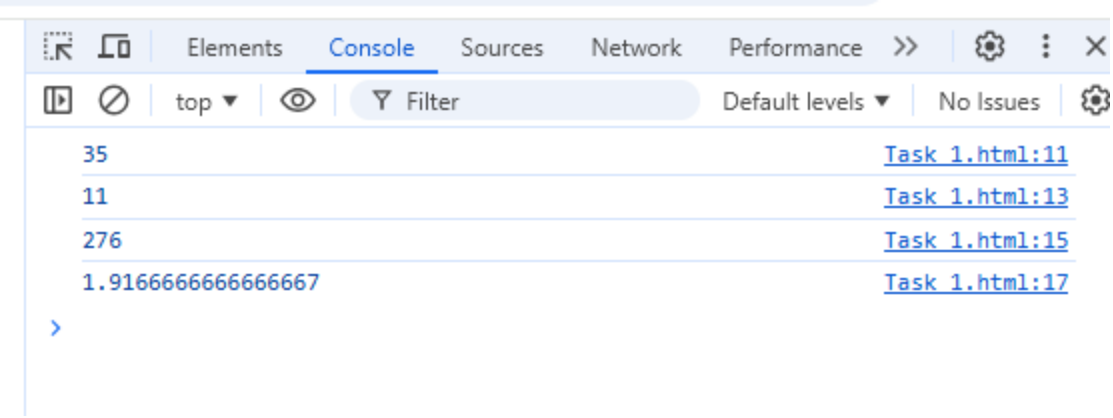
OUTPUT:



TASK 3:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        var num1=23;
        var num2=12;
        var add=num1+num2;
        console.log(add);
        var sub=num1-num2;
        console.log(sub);
        var mul=num1*num2;
        console.log(mul);
        var div=num1/num2;
        console.log(div);
    </script>
</html>
```

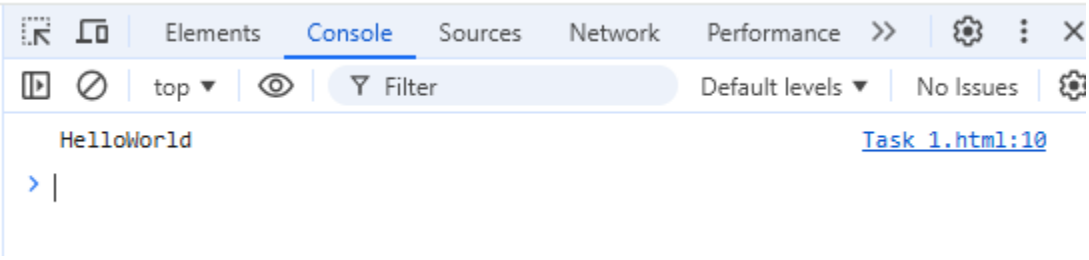
OUTPUT:



TASK 4:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        var str1="Hello";
        var str2="World";
        console.log(str1+str2);
    </script>
</html>
```

OUTPUT:

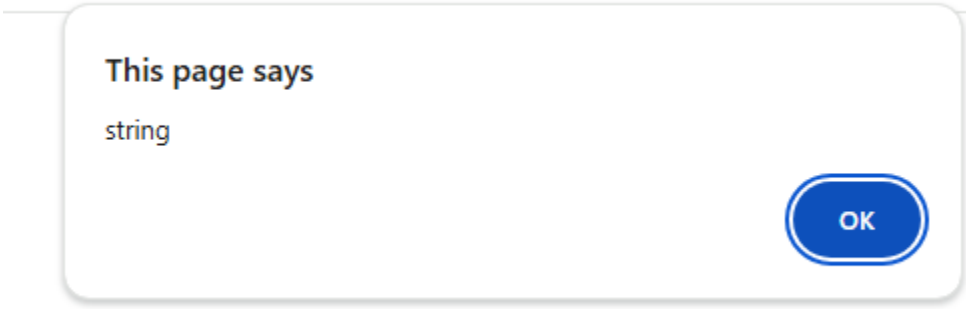


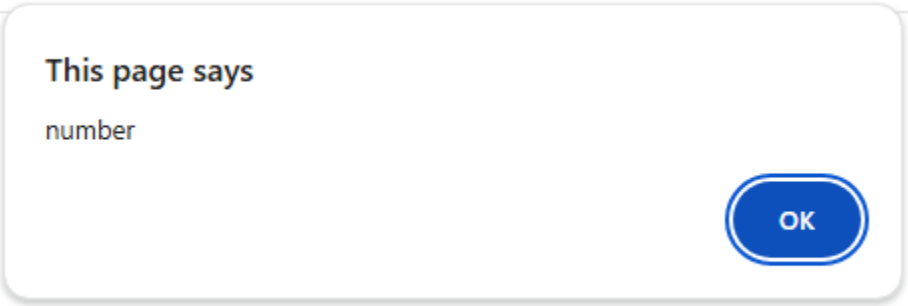
TASK 5:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>Task 1</title>
    <script>
        var name="john";
        alert(typeof name);
        let age=12;
        alert(typeof age);
    </script>
</html>
```

OUTPUT:





TASK 6:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        /*
        This is a multi-line comment.
        Hello! This is Parameshwari.
        */
        // This is a single-line comment.
        I am learning MernStack.
    </script>
</html>
```

Explanation of the Difference:

- **Multi-line comment**:
  - Begins with `/*` and ends with `*/`.
  - It can span across multiple lines, making it useful for longer descriptions or temporarily commenting out large sections of code.
- **Single-line comment**:
  - Begins with `//` and only applies to the text following it on that line.
  - It is often used for brief comments or explanations within the code, typically on a single line.

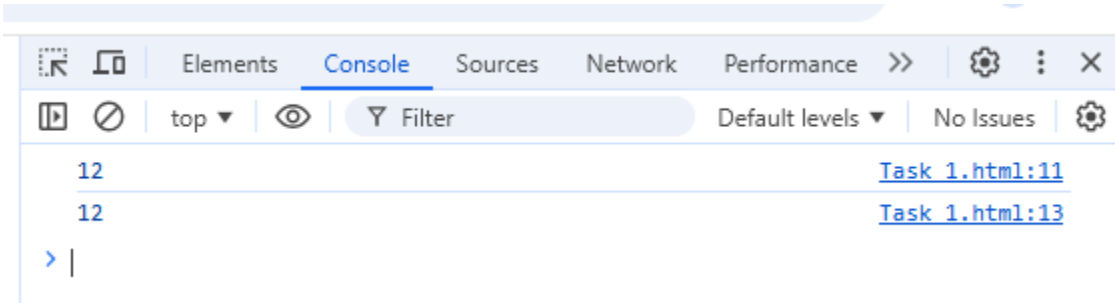
TASK 7:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        let x=5;
        let y=7;
        let sum=x+y;
        console.log(sum);
        let total=x+y
        console.log(total)
    </script>
</html>
```

Differences in Behavior:

- **Semicolon-separated**: The behavior is explicit, and there are no ambiguities. Each statement is clearly separated.
- **Non-semicolon-separated**: JavaScript uses ASI to automatically insert semicolons at the end of lines, but this can sometimes lead to unexpected behavior, especially in complex code. For example, in some cases (like when returning values from a function), ASI might not work as expected and cause errors.

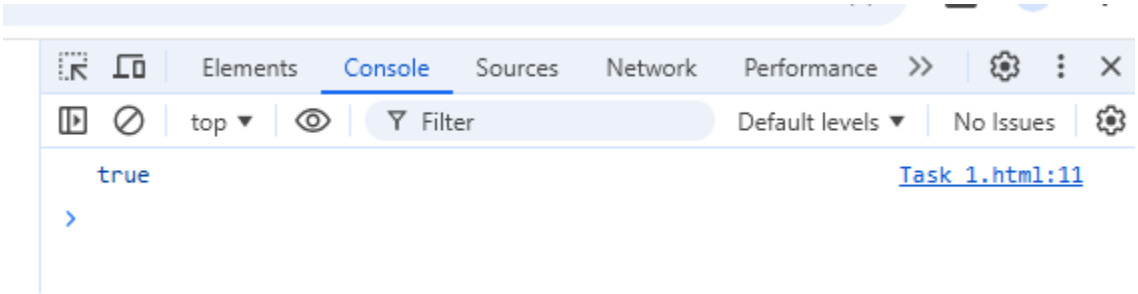
OUTPUT:



TASK 8:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        var boolean=true;
        if(boolean)
        {
            console.log(boolean);
        }
    </script>
</html>
```

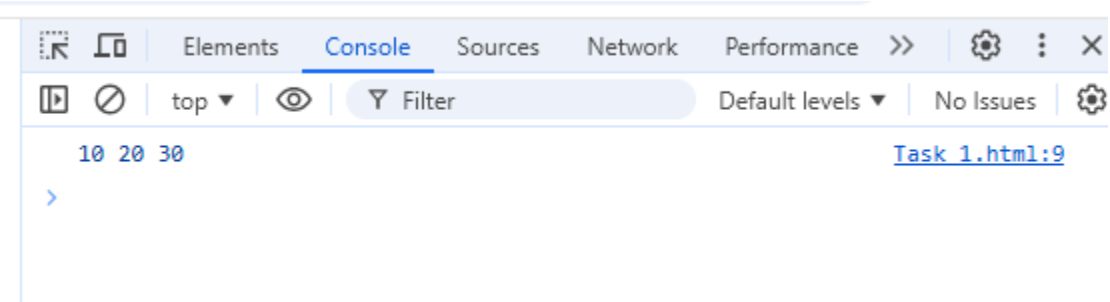
OUTPUT:



TASK 9:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 1</title>
    <script>
        let x=10,y=20,z=30;
        console.log(x,y,z);
    </script>
</html>
```

OUTPUT:



TASK 10:

Scenario 1: `<script>` at the top of the document (inside the `<head>` section)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Script at Top</title>
    <script>
        console.log('Script at the top of the document');
    </script>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>This is an example of a script tag at the top.</p>
</body>
</html>
```

Scenario 2: `<script>` at the bottom of the document (just before `</body>`)

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Script at Bottom</title>
</head>
<body>
    <h1>Hello, World!</h1>
    <p>This is an example of a script tag at the bottom.</p>

    <script>
        console.log('Script at the bottom of the document');
    </script>
</body>
</html>
```

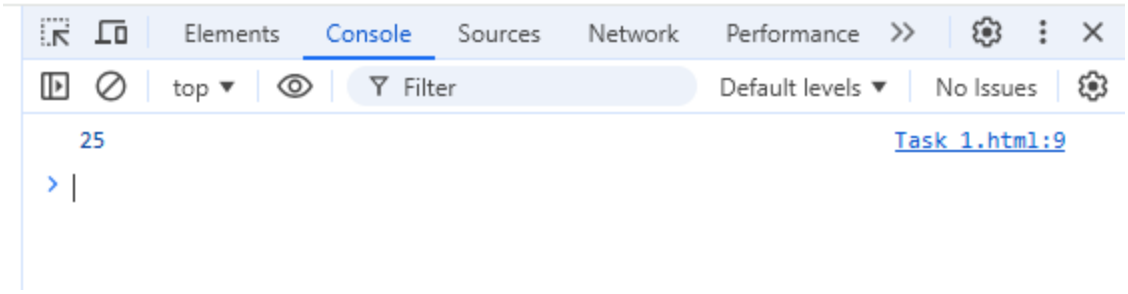
Key Differences in Behavior:

- **Execution Timing:**
  - **Top of the document**: The script is executed before the content is rendered, potentially blocking the rendering.
  - **Bottom of the document**: The script is executed after the HTML content has been parsed and rendered.
- **Page Load Speed:**
  - **Top of the document**: Can slow down the initial rendering of the page.
  - **Bottom of the document**: Faster rendering, as the script is fetched and executed after the page is displayed.

TASK 11:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 11</title>
    <script>
        num = 25;
        console.log(num);
    </script>
</head>
</html>
```

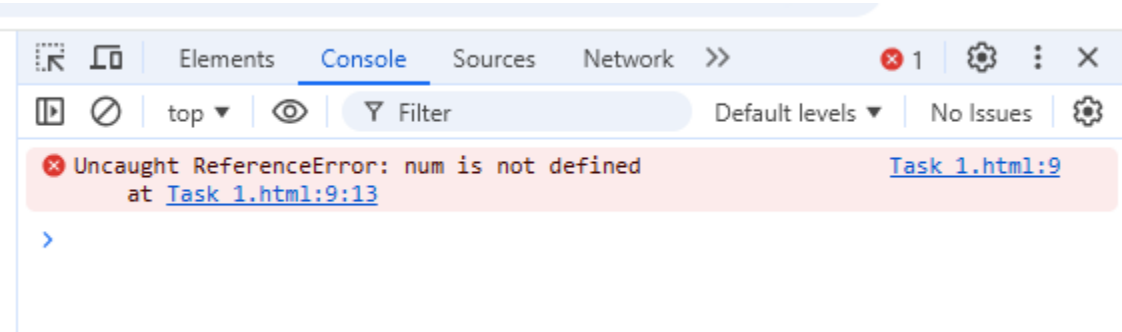
OUTPUT:



TASK 12:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 12</title>
    <script>
        "use strict";
        num = 25;
        console.log(num);
    </script>
</head>
</html>
```

OUTPUT:

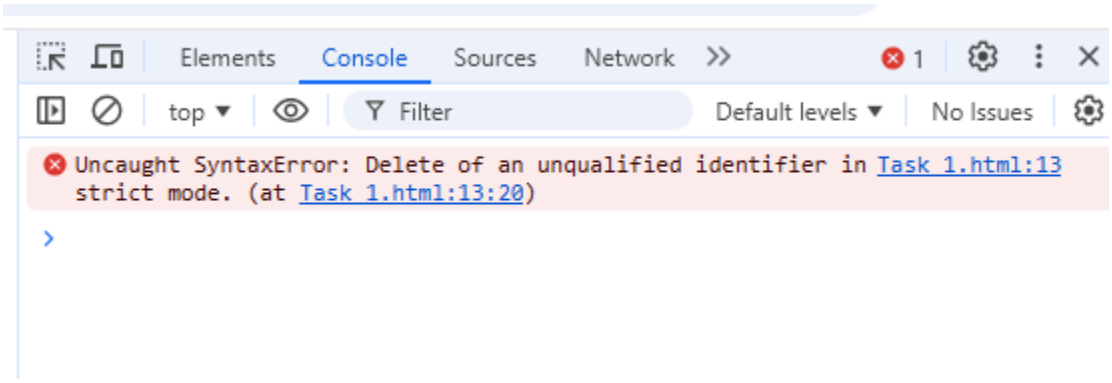


TASK 13:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test</title>
</head>
<body>
    <script>
        "use strict";
        var myVar = 10;
        try {
            delete myVar;
        } catch (e) {
            console.log("Error deleting variable: " + e.message);
        }
        function myFunction() {
            return "Hello!";
        }
        try {
            delete myFunction;
        } catch (e) {
            console.log("Error deleting function: " + e.message);
        }
        function myFunctionWithParam(param) {
            try {
                delete param;
            } catch (e) {
                console.log("Error deleting function parameter: " + e.message);
            }
        }
        myFunctionWithParam("test");
```

```
    </script>
</body>
</html>
```

OUTPUT:

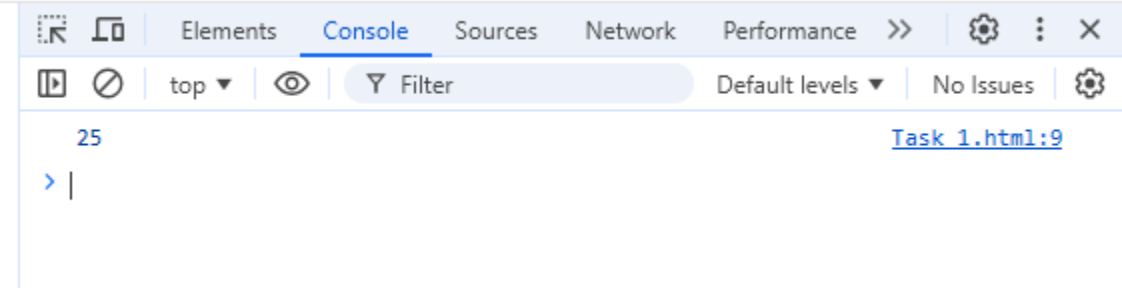


TASK 14:

WITHOUT USE STRICT:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test</title>
</head>
<body>
    <script>
       num = 25;
       console.log(num);
    </script>
</body>
</html>
```

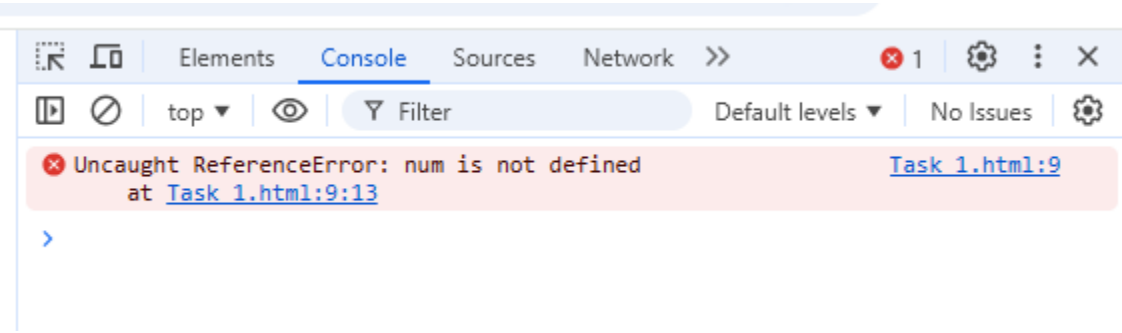
OUTPUT:



WITH USE STRICT:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task 12</title>
    <script>
        "use strict";
        num = 25;
        console.log(num);
    </script>
</head>
</html>
```

OUTPUT:

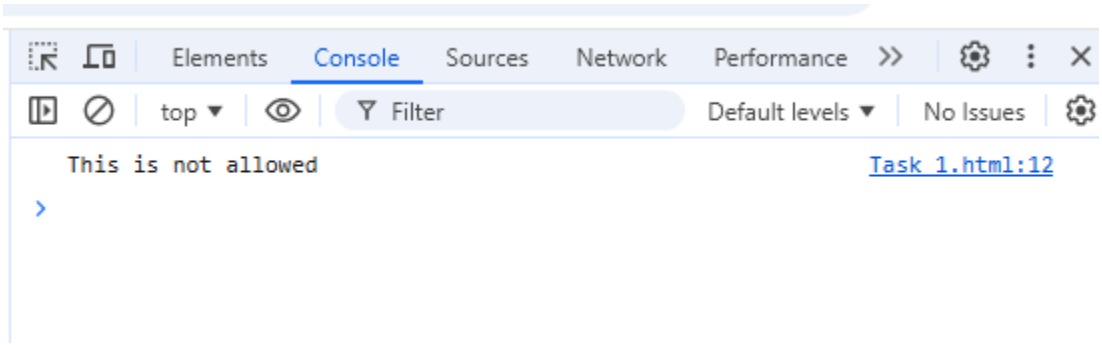


TASK 15:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
```

```
<body>
    <script>
        "use strict";
        var name = "This is not allowed";
        console.log(name);
    </script>
</body>
</html>
```

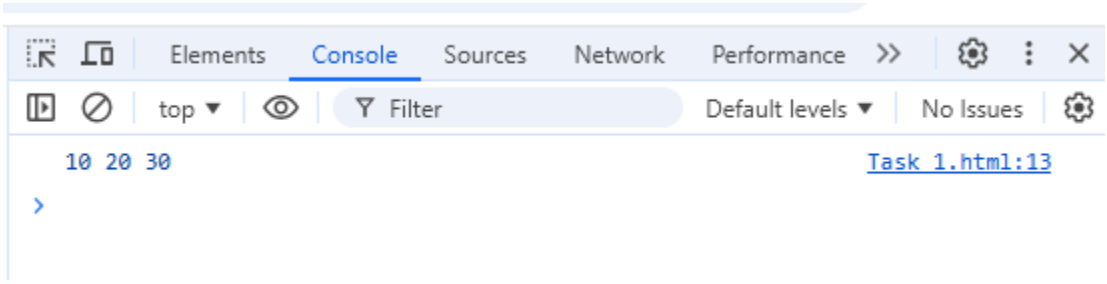
OUTPUT:



TASK 16:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
<body>
    <script>
        var x = 10;
        let y = 20;
        const z = 30;
        console.log(x,y,z);
    </script>
</body>
</html>
```

OUTPUT:



- **Use `const` by default**: Most variables should be constants. If the variable's value is not going to change, `const` is a good choice because it ensures immutability.

- **Use `let` when reassignment is necessary**: If you need to reassign the variable, such as in loops or conditional statements, use `let`.

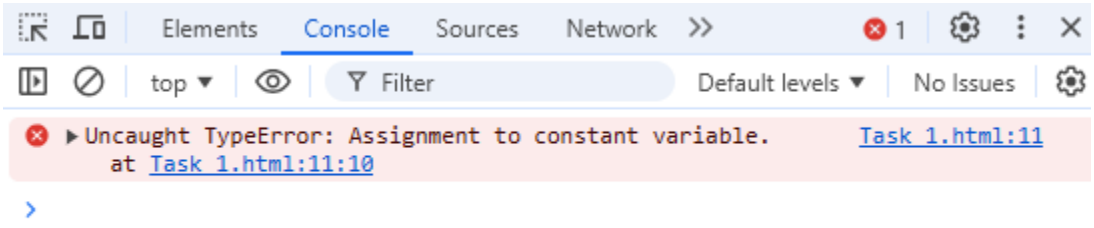
- **Avoid `var`**: In modern JavaScript, `var` is generally avoided because its function-scoping and hoisting behavior can lead to bugs. Stick to `let` and `const` for better scoping and cleaner, more predictable code.

TASK 17:

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
<body>
    <script>
        const z = 30;
        z=20;
        console.log(z);
    </script>
</body>
</html>
```

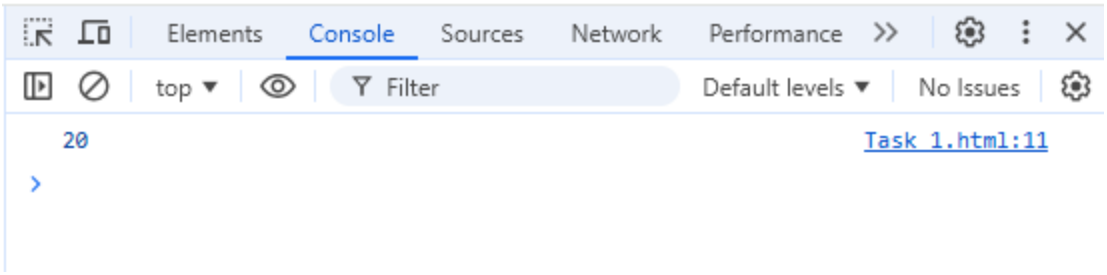
OUTPUT:



TASK 18:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
<body>
    <script>
        z=20;
        console.log(z);
    </script>
</body>
</html>
```

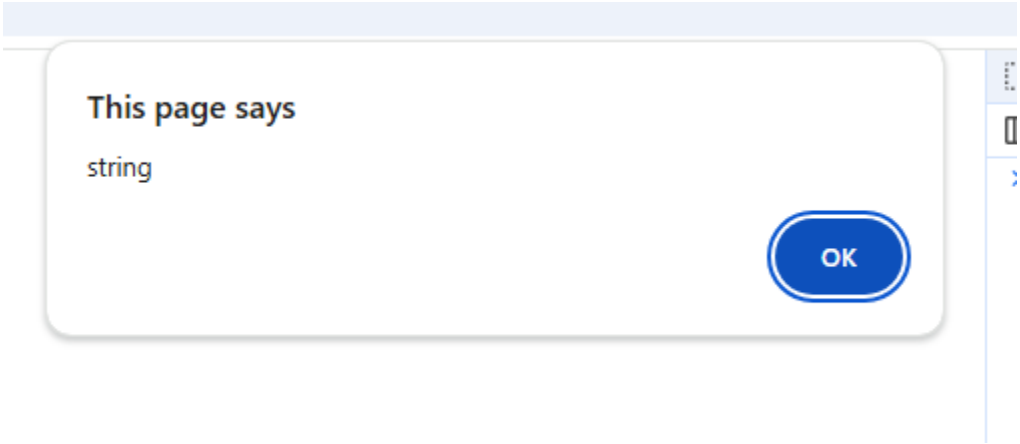
OUTPUT:

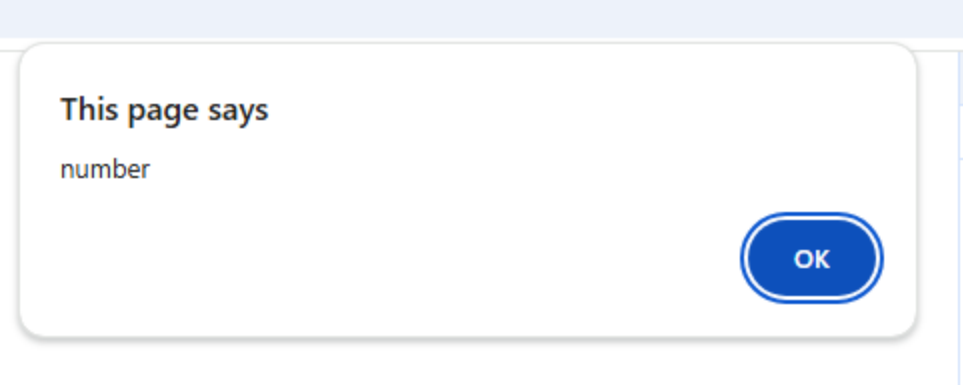


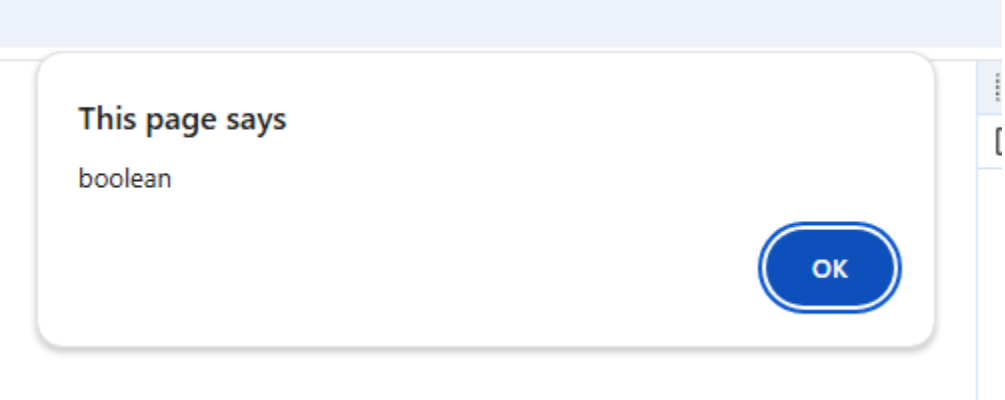
TASK 19:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
<body>
    <script>
        var name="Johny";
        let age=19;
        var boolean=false;
        alert(typeof name);
        alert(typeof age);
        alert(typeof boolean);
    </script>
</body>
</html>
```

OUTPUT:



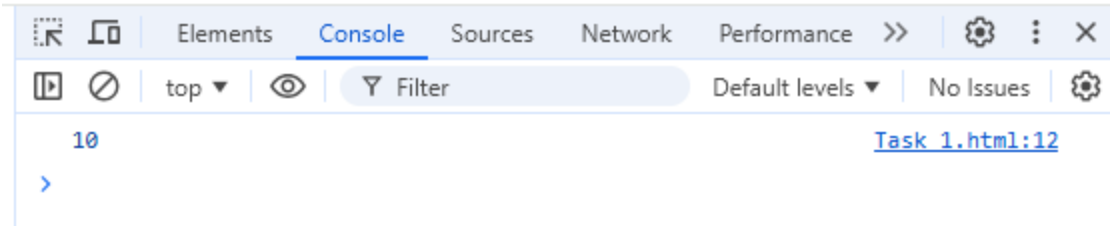




TASK 20:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>"use strict" Test - Reserved Keyword</title>
</head>
<body>
    <script>
        let a=10;
        let b=a;
        console.log(b);
    </script>
</body>
</html>
```

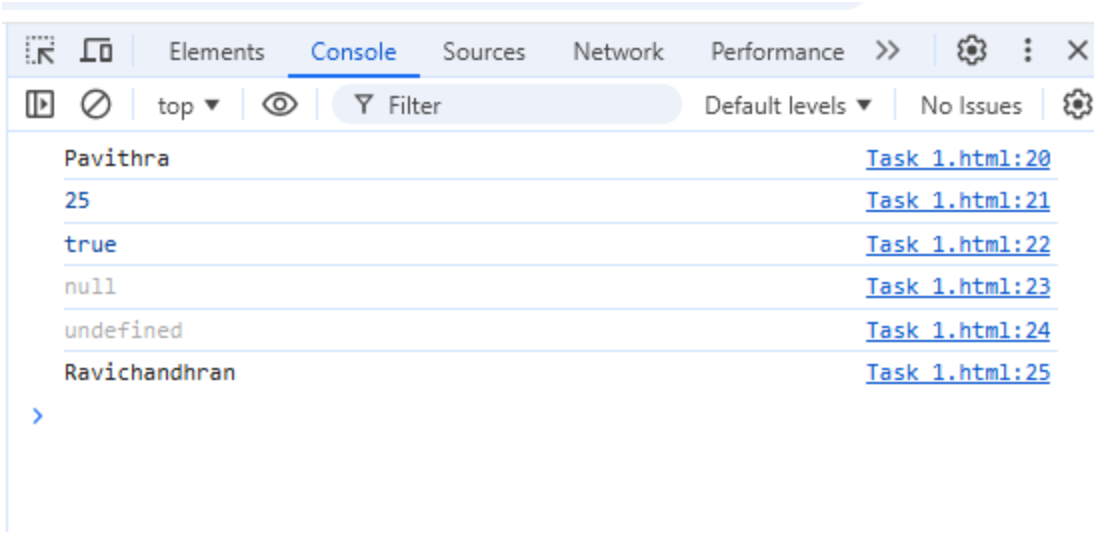
OUTPUT:


Elements
Console
Sources
Network
Performance
top
Filter
Default levels
No Issues
10
Task 1.html:12

TASK 21:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Types in JavaScript</title>
</head>
<body>
    <script>
        let name = "Pavithra";
        let age = 25;
        let isActive = true;
        let define = null;
        let notAssigned;  /
        let person = {
            firstName: "Pavithra",
            lastName: "Ravichandhran",
            age: 25
        };
        console.log(name);
        console.log(age);
        console.log(isActive);
        console.log(define);
        console.log(notAssigned);
        console.log(person.lastName);
    </script>
</body>
</html>
```

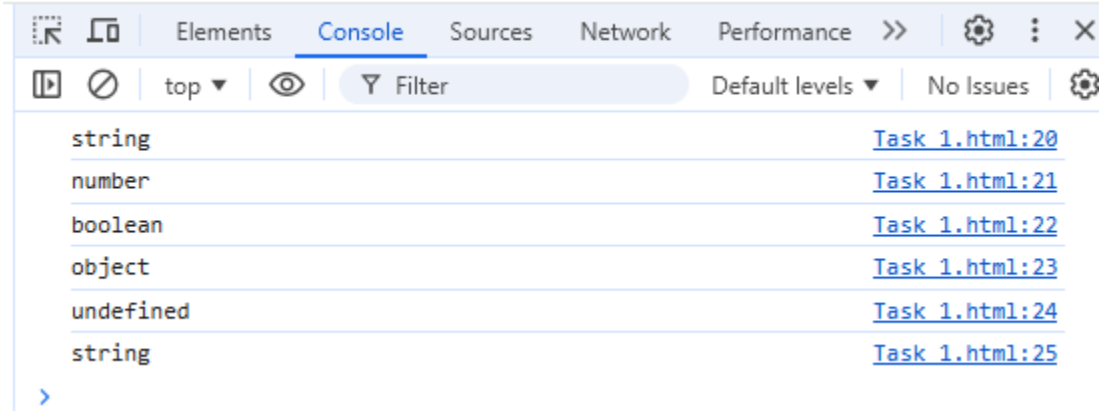
OUTPUT:



TASK 22:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Types in JavaScript</title>
</head>
<body>
    <script>
        let name = "Pavithra";
        let age = 25;
        let isActive = true;
        let define = null;
        let notAssigned;
        let person = {
            firstName: "Pavithra",
            lastName: "Ravichandhran",
            age: 25
        };
        console.log(typeof name);
        console.log(typeof age);
        console.log(typeof isActive);
        console.log(typeof define);
        console.log(typeof notAssigned);
        console.log(typeof person.lastName);
    </script>
</body>
</html>
```

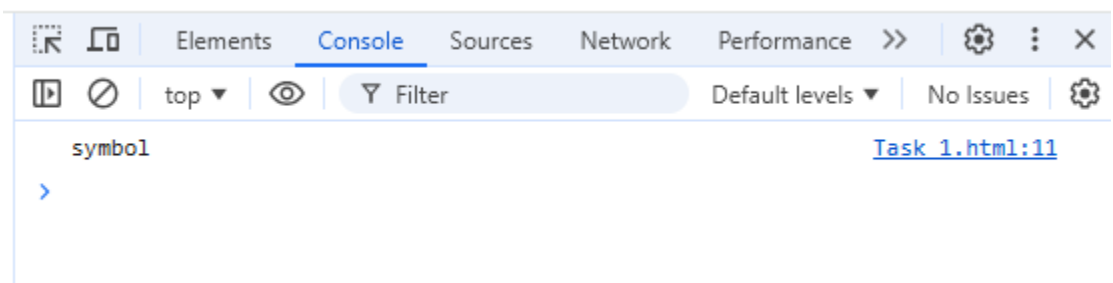
OUTPUT:



TASK 23:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Types in JavaScript</title>
</head>
<body>
    <script>
        let name = Symbol("Nivi");
        console.log(typeof name);
    </script>
</body>
</html>
```

OUTPUT:

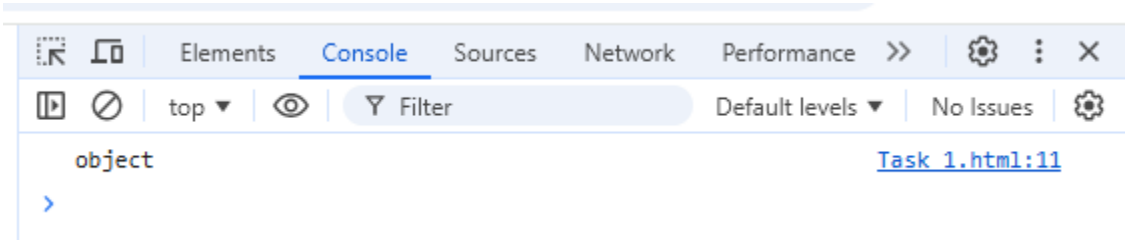


TASK 24:

```
<!DOCTYPE html>
```

```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Data Types in JavaScript</title>
</head>
<body>
    <script>
        let name = null;
        console.log(typeof name);
    </script>
</body>
</html>
```

OUTPUT:

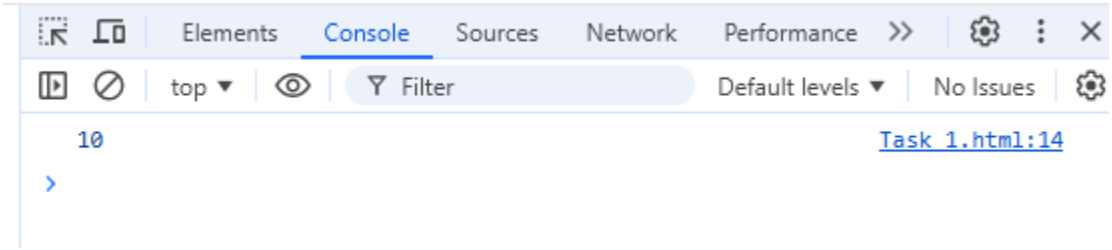


TASK 25:

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        function varScopeExample() {
            if (true) {
                var x = 10;
            }
            console.log(x);
        }
        function letScopeExample() {
            if (true) {
                let y = 20;
            }
        }
        varScopeExample();
```

```
        letScopeExample();
    </script>
</body>
</html>
```

OUTPUT:

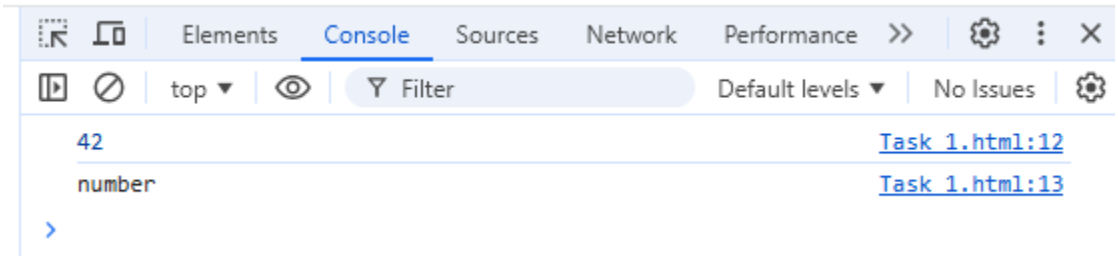


TASK 26:

Implicit Conversion (Type Coercion)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        let str = "42";
        let result = str * 1;
        console.log(result);
        console.log(typeof result);
    </script>
</body>
</html>
```

OUTPUT:

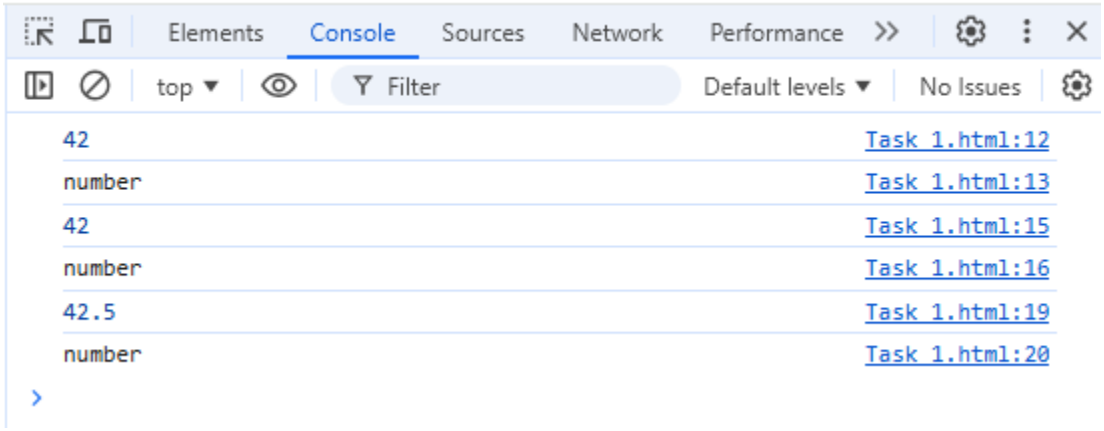


Explicit Conversion (Using Built-in Functions)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        let str = "42";
        let num1 = Number(str);
        console.log(num1);
        console.log(typeof num1);
```

```
        let num2 = parseInt(str);
        console.log(num2);
        console.log(typeof num2);
        let str2 = "42.5";
        let num3 = parseFloat(str2);
        console.log(num3);
        console.log(typeof num3);
    </script>
</body>
</html>
```

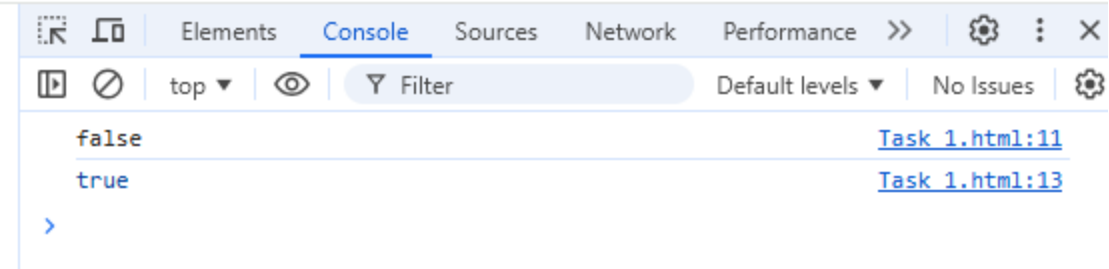
OUTPUT:



TASK 27:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        var boolean=false;
        console.log(String(boolean));
    </script>
</body>
</html>
```

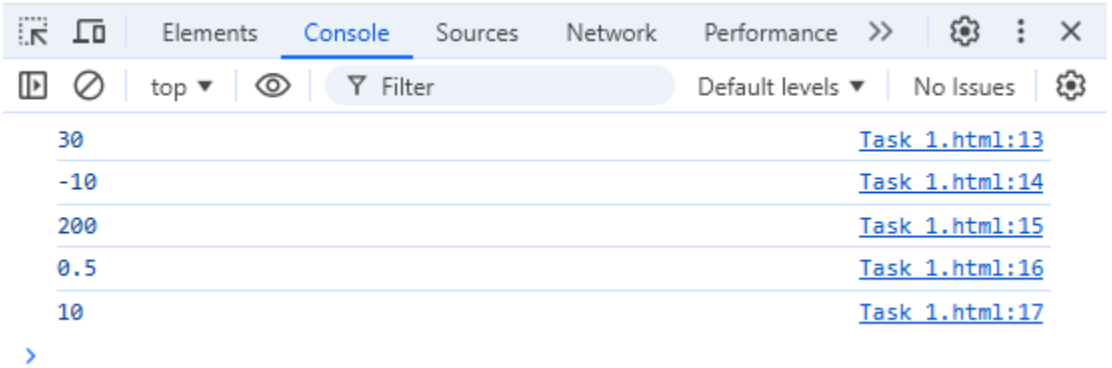
OUTPUT:



TASK 28:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        var a=10;
        var b=20;
        var add=a+b;
        console.log(add);
        console.log(a-b);
        console.log(a*b);
        console.log(a/b);
        console.log(a%b);
    </script>
</body>
</html>
```

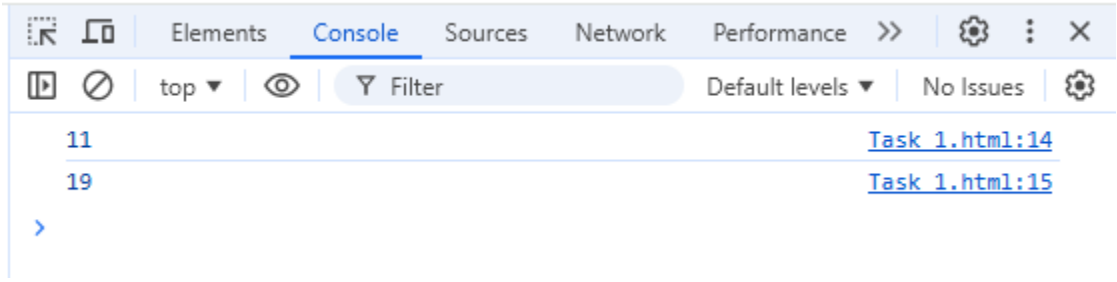
OUTPUT:



TASK 29:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        var a=10;
        var b=20;
        a++;
        b--;
        console.log(a);
        console.log(b);
    </script>
</body>
</html>
```

OUTPUT:

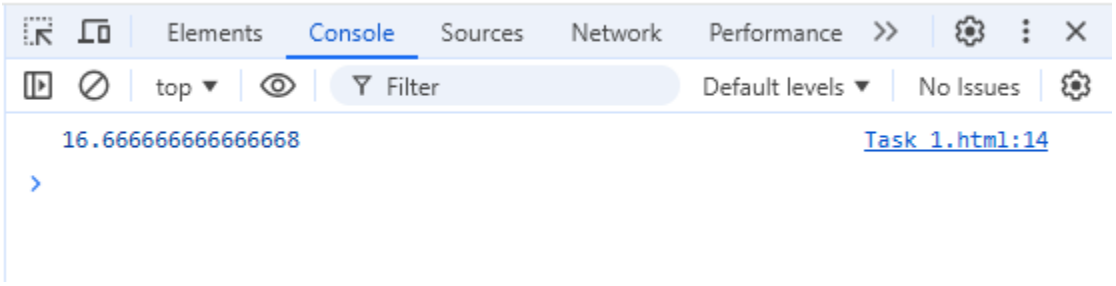


TASK 30:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>var vs let Scope</title>
</head>
<body>
    <script>
        let a=10;
        let b=20;
        let c=30;
```

```
        let fun=a+(b*a)/c;
        console.log(fun);
    </script>
</body>
</html>
```

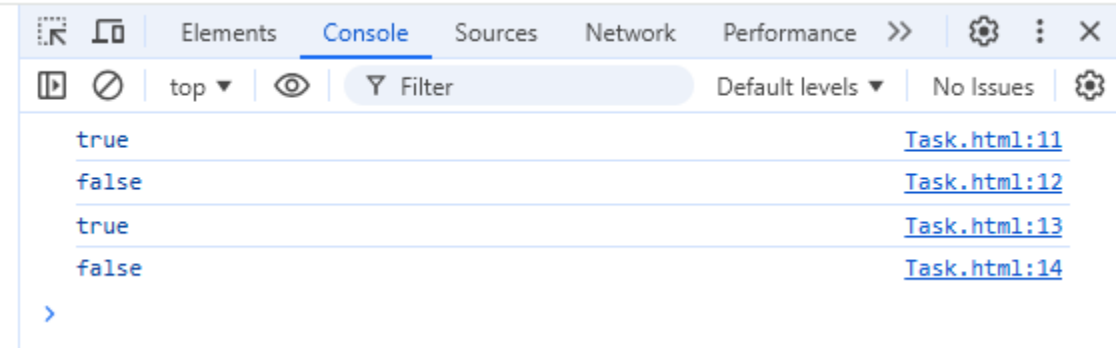
OUTPUT:



TASK 31:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a = 7;
        let b = 5;
        console.log( a > b);
        console.log( a < b);
        console.log( a >= b);
        console.log( a <= b);
    </script>
</html>
```

OUTPUT:

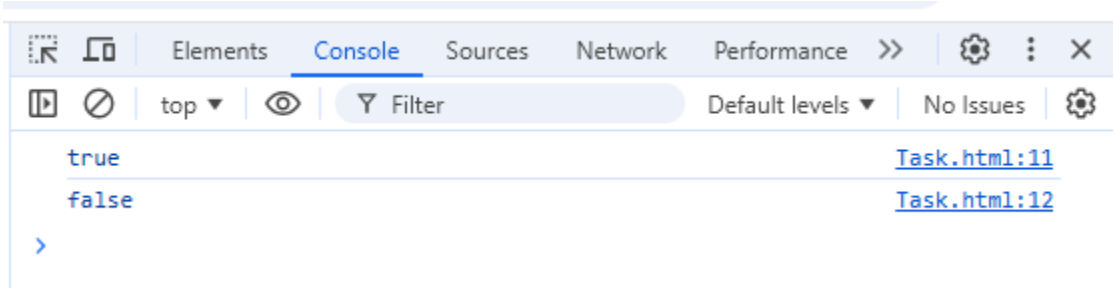


TASK 32:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
```

```
    </head>
    <script>
        let a="5";
        let b=5;
        console.log( a == b);
        console.log( a === b);
    </script>
</html>
```

OUTPUT:



Key Differences:

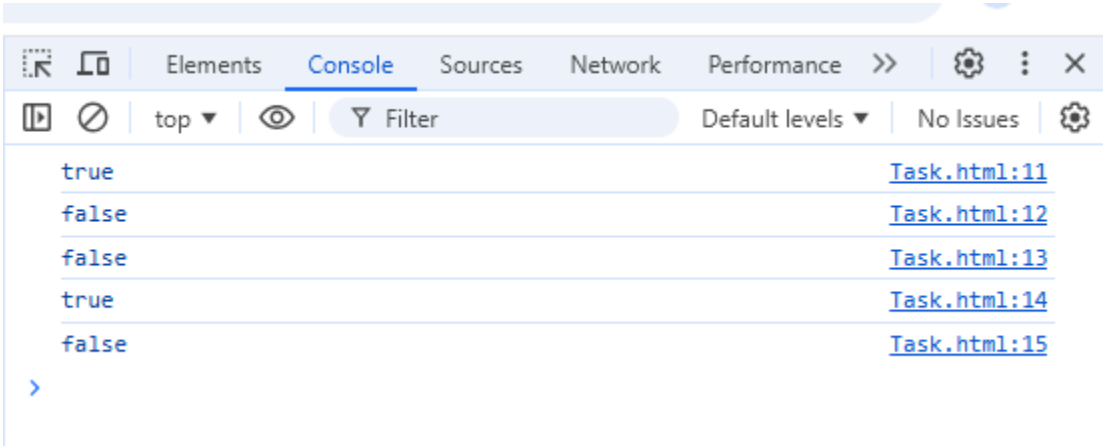
- **Equality (==)**: Performs type coercion. For example, `num == str` is `true` because JavaScript converts the string `"5"` to the number `5` before comparing.
- **Strict Equality (===)**: Does **not** perform type coercion. For example, `num === str` is `false` because one is a number and the other is a string.

TASK 33:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
```

```
    <script>
        let str1 = "apple";
        let str2 = "banana";
        console.log(str1 < str2);
        console.log(str1 > str2);
        console.log(str1 === str2);
        console.log(str1 <= str2);
        console.log(str1 >= str2);
    </script>
</html>
```

OUTPUT:

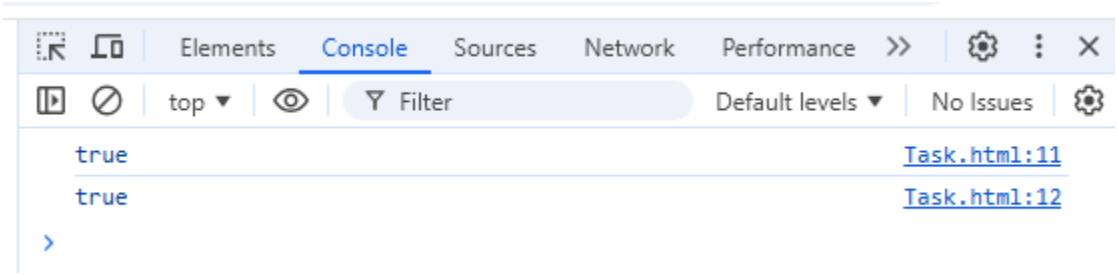


TASK 34:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
```

```
        let a=5;
        let b=10;
        console.log(a!=b);
        console.log(a!==b);
    </script>
</html>
```

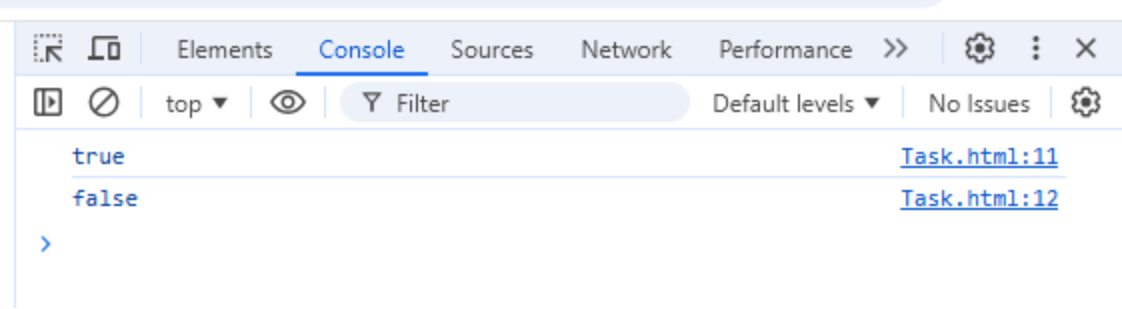
OUTPUT:



TASK 35:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=null;
        let b;
        console.log(a==b);
        console.log(a===b);
    </script>
</html>
```

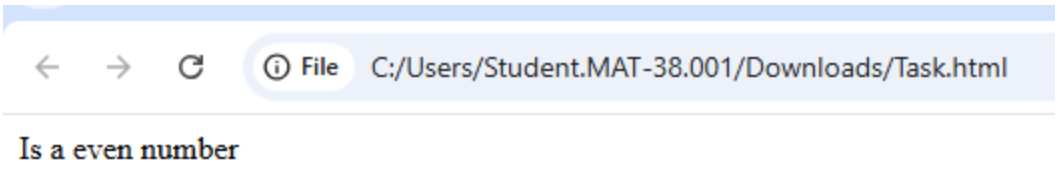
OUTPUT:



TASK 36:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let num=6;
        if(num%2==0)
        {
            document.writeln("Is a even number");
        }
        else
        {
            document.writeln("Is a odd number");
        }
    </script>
</html>
```

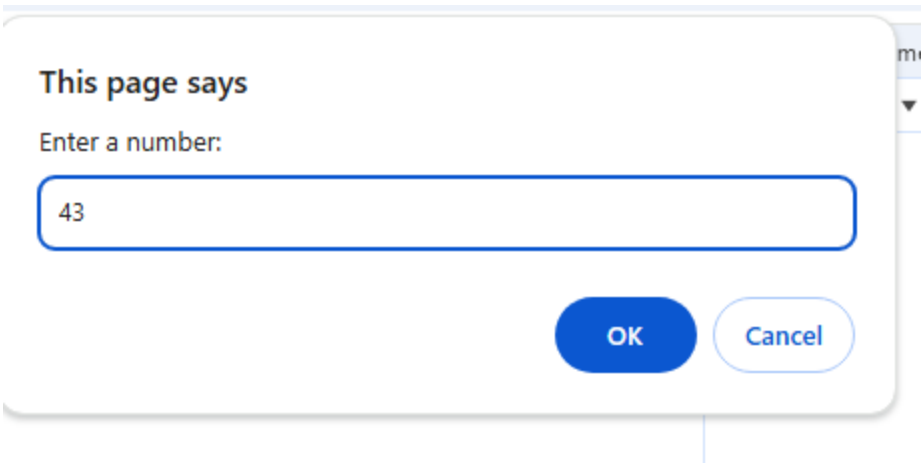
OUTPUT:

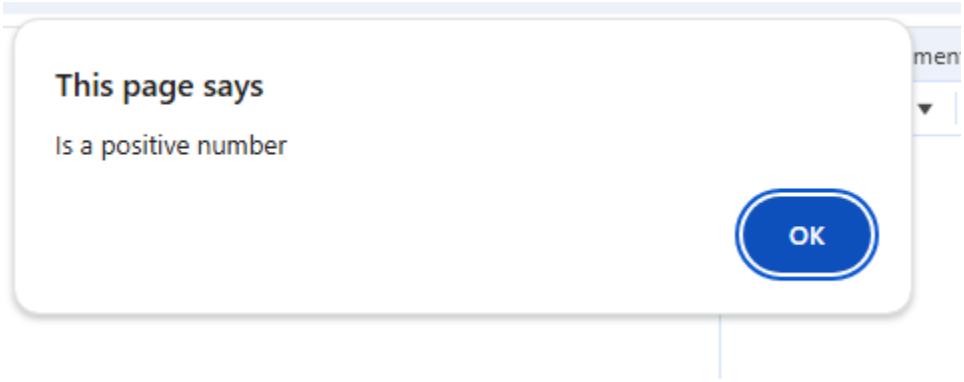


TASK 37:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        var num=prompt("Enter a number:");
        if(num==0)
        {
            alert("Is a zero");
        }
        else if(num<0)
        {
            alert("Is a negative number");
        }
        else
        {
            alert("Is a positive number");
        }
    </script>
</html>
```

OUTPUT:

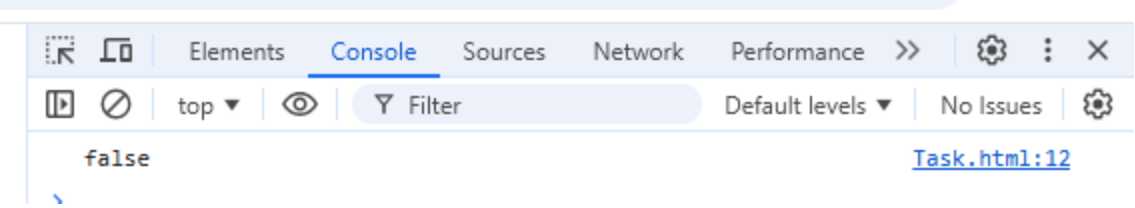




TASK 38:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=10;
        let b=5;
        let result=(a<b?"true":"false");
        console.log(result);
    </script>
</html>
```

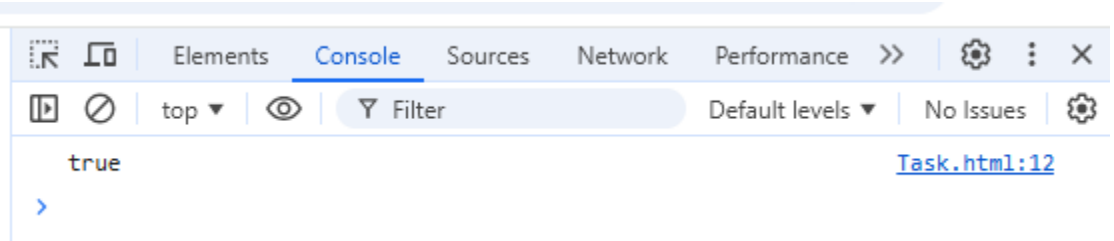
OUTPUT:



TASK 39:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=10;
        let b=5;
        let result=(a/b?"true":"false");
        console.log(result);
    </script>
</html>
```

OUTPUT:

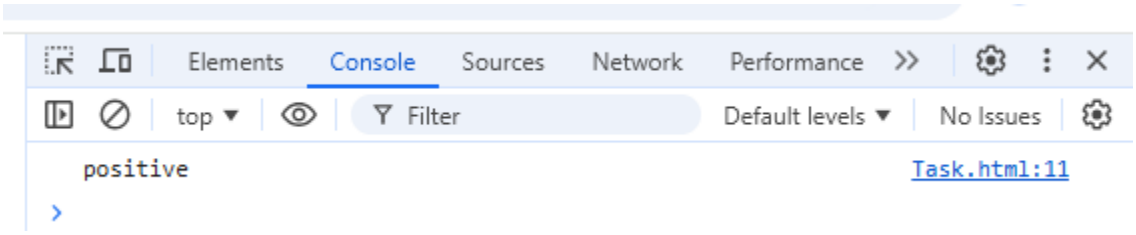


TASK 40:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
```

```
    </head>
    <script>
        let num=8
        let result=(num>0?"positive":"negative");
        console.log(result);
    </script>
</html>
```

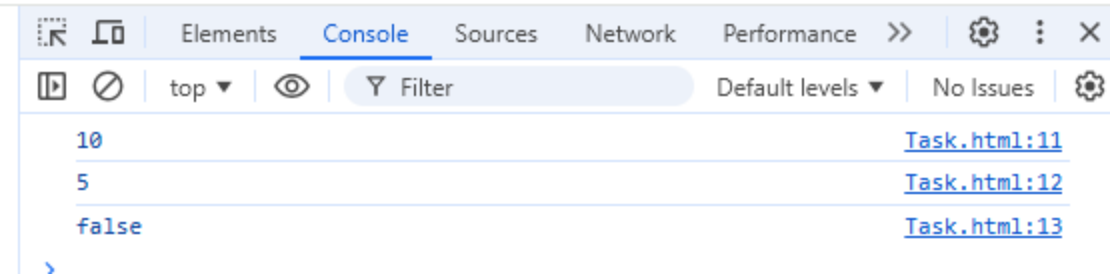
OUTPUT:



TASK 41:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let num1=5;
        let num2=10;
        console.log(num1&&num2);
        console.log(num1||num2);
        console.log(!num1);
    </script>
</html>
```

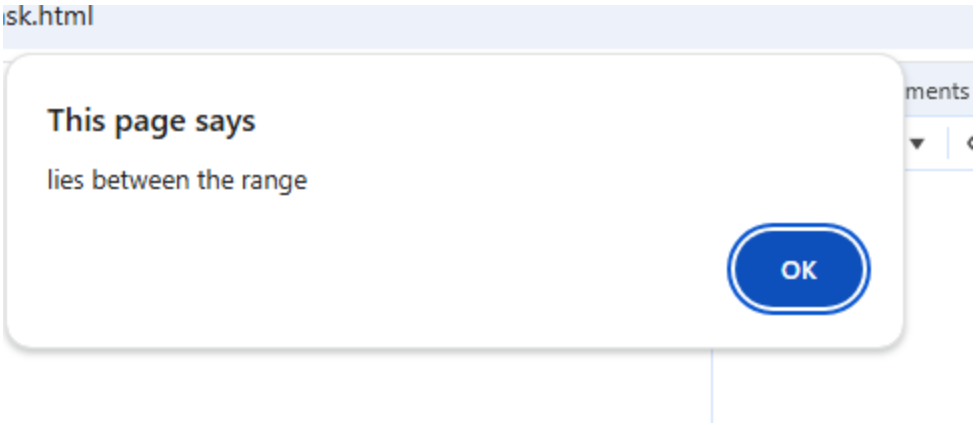
OUTPUT:



TASK 42:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=5;
        if(a>=0 &&  a<=20)
    {
        alert("lies between the range");
    }
    else{
        alert("not lies");
    }
    </script>
</html>
```

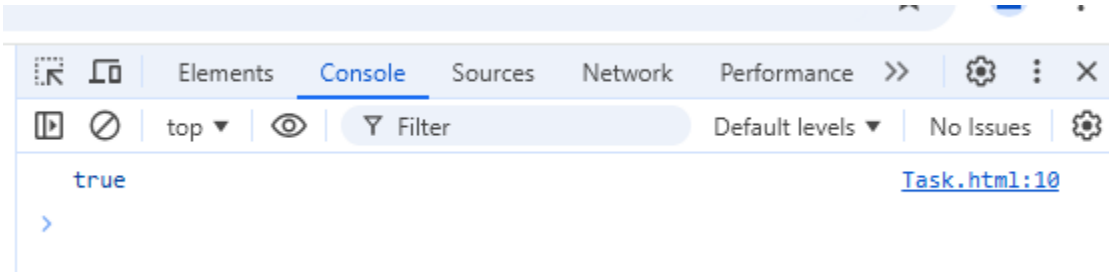
OUTPUT:



TASK 43:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=false;
        console.log(!a);
    </script>
</html>
```

OUTPUT:

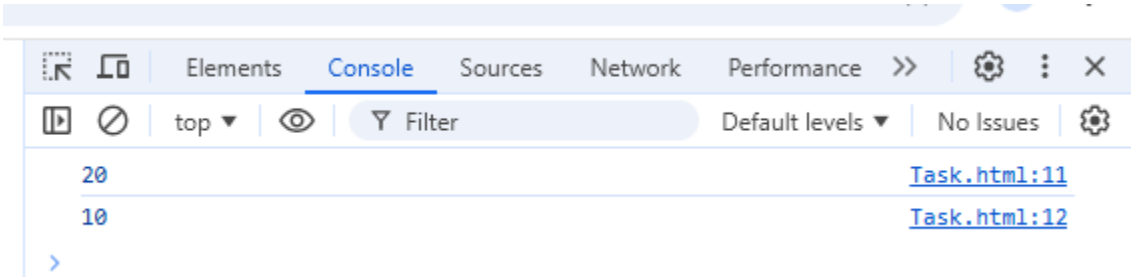


TASK 44:

```
<!DOCTYPE html>
<html>
```

```
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a=10;
        let b=20;
        console.log(a&&b);
        console.log(a||b);
    </script>
</html>
```

OUTPUT:

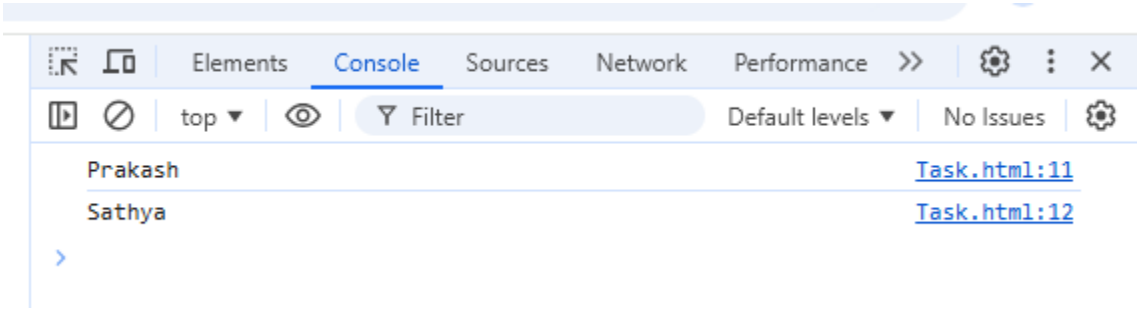


TASK 45:

```
<!DOCTYPE html>
<html>
   <head>
        <meta charset="UTF-8">
        <meta name:"viewport" content="width=device-
width,initial-scale=1.0">
        <title>Task</title>
    </head>
    <script>
        let a="Sathya";
        let b="Prakash";
        console.log(a&&b);
```

```
        console.log(a||b);
    </script>
</html>
```

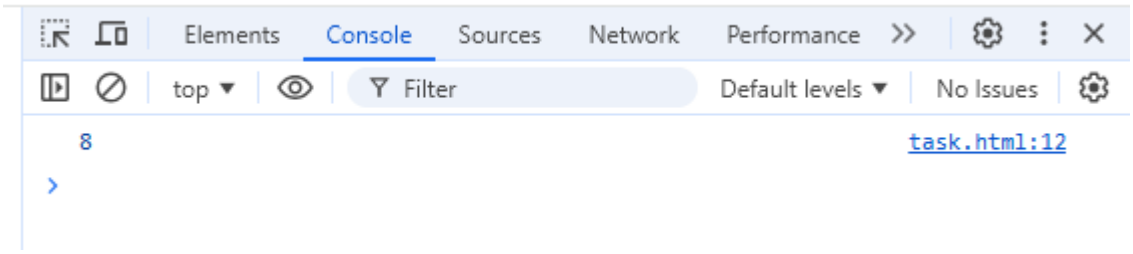
OUTPUT:



TASK 46:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<script>
   function add(a,b)
   {
      return a+b;
   }
   console.log (add(5,3));
</script>
</body>
</html>
```

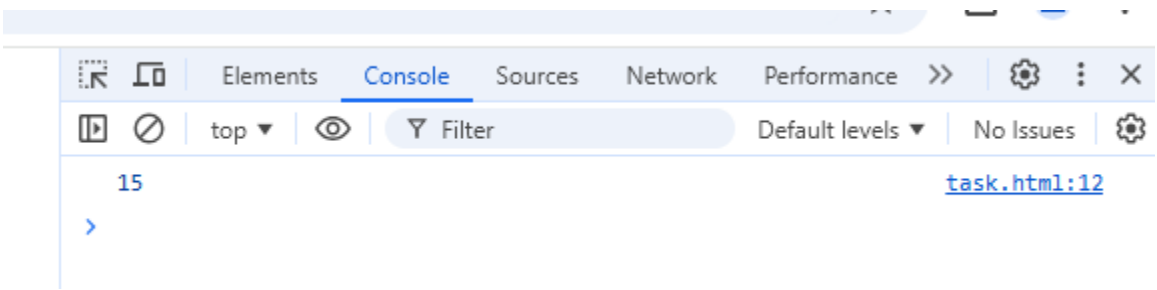
OUTPUT:



TASK 47:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<script>
   function area(a,b)
   {
      return a*b;
   }
   console.log (area(5,3));
</script>
</body>
</html>
```

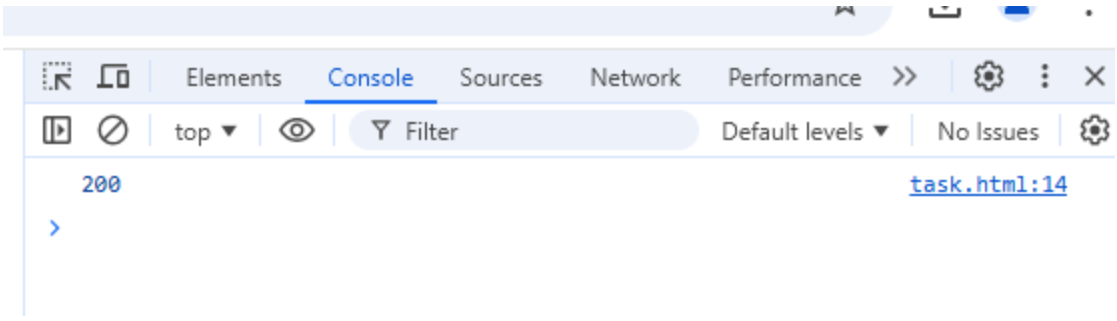
OUTPUT:



TASK 48:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<script>
   function parameter()
   {
      let a=10;
      let b=20;
      return a*b;
   }
   console.log (parameter());
</script>
</body>
</html>
```

OUTPUT:

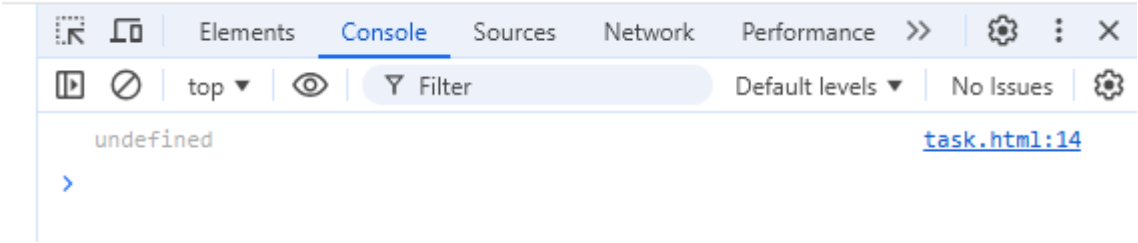


TASK 49:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
</head>
<script>
   function parameter()
   {
      let a=10;
      let b=20;
      return ;
   }
   console.log (parameter());
</script>
</body>
</html>
```

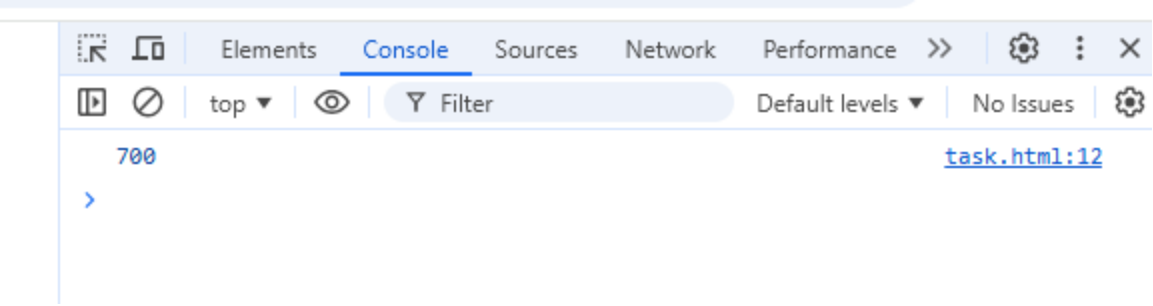
OUTPUT:



TASK 50:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<script>
   function parameter(a,b=20)
   {
      return a*b;
   }
   console.log (parameter(35));
</script>
</body>
</html>
```

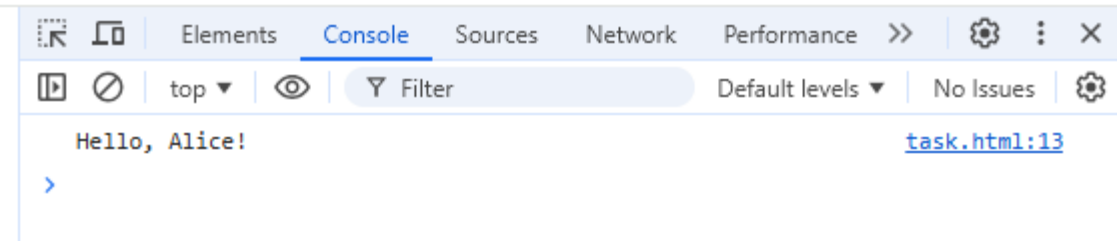
OUTPUT:



TASK 51:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<script>
  let greet = (name) => {
    return `Hello, ${name}!`;
  }
  console.log(greet("Alice"));
</script>
</body>
</html>
```

OUTPUT:

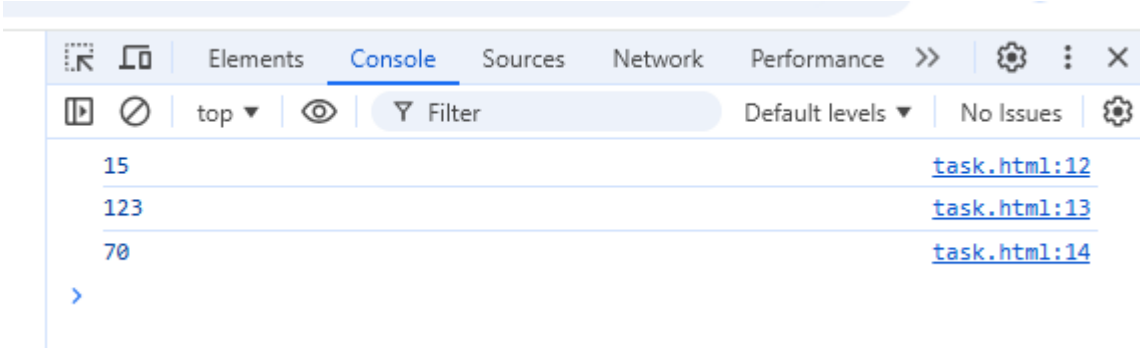


TASK 52:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<script>
```

```
    let add = (num1,num2) => {
        return num1+num2;
    }
    console.log(add(7,8));
    console.log(add(78,45));
    console.log(add(37,33));
</script>
</body>
</html>
```

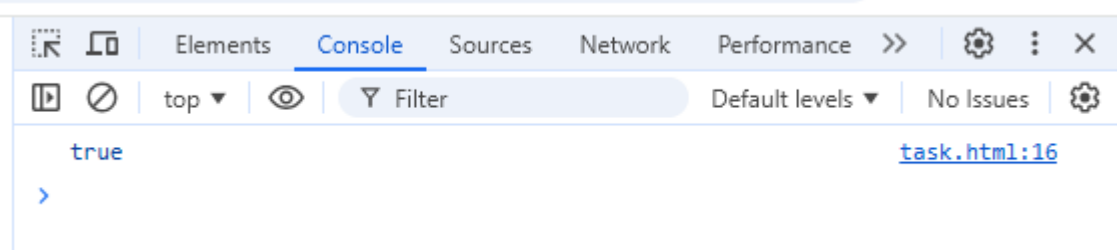
OUTPUT:



TASK 53:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<script>
    let even=(num)=>
{
    if(num%2==0)
        return true;
    else
        return false;
}
    console.log(even(8));
</script>
</body>
</html>
```

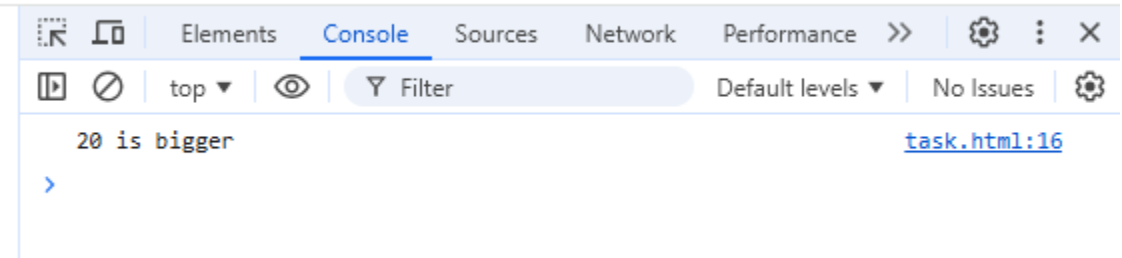
OUTPUT:



TASK 54:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<script>
   let maxValue = (a, b) => {
      if (a > b) {
         return `${a} is bigger`;
      } else {
         return `${b} is bigger`;
      }
   };
   console.log(maxValue(8, 20));
</script>
</body>
</html>
```

OUTPUT:



TASK 55:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
    <title>this Behavior</title>
</head>
<body>
<script>
    let myObject = {
        value: 10,
        multiplyTraditional: function(num) {
            return this.value * num;
        },
        multiplyArrow: (num) => {
            return this.value * num;
        }
    };
    console.log(myObject.multiplyTraditional(5));
    console.log(myObject.multiplyArrow(7));
</script>
</body>
</html>
```

OUTPUT: